

明 細 書

リアルタイム制御システム

技術分野

本発明はリアルタイム制御システムに係り、特に、周期的なイベントを検出し、又は複数の信号のイベントを検出してタスク処理を行うリアルタイム制御システムに関する。

背景技術

自動車の制御コントローラ及びその他の産業機器の制御コントローラにおいて、制御対象からの信号を受信してから処理を開始するまでの応答性能を向上させ、かつ制約時間内に必要な処理を確実に実行し終了させるような、リアルタイム性が求められている。特に、自動車の制御応答性及び安全性の確保などを課題として所定の処理をリアルタイムに実行する必要があるオペレーティングシステム（以下OSと云う）においては、周期的にタスクを起動する周期タスクと、イベント発生時にタスクを起動するイベントタスクと、をサポートする必要がある。

例えば、非特許文献であるOSEK/VDX Operating System Specification 2.2.1に示される如きような自動車に搭載された標準的なOSは、タイムテーブルサービスという、周期的に複数のタスクを起動する機能を提供している。このタイムテーブルサービスを開始するシステムコールを実行すると、登録されたタスク群が、指定された周期毎に起動され、この起動により、周期タスクが、実現される。

このような、OSを有したリアルタイム制御システムは、自動車のエンジン制御用のコントローラで用いられており、このコントローラは、周期タスクを用いて、時間駆動で動作するようにシステムが構築されることが多い。なぜなら、こ

の方式では、タスクの実行間隔を調整することにより、演算負荷を調節しやすいことと、タスクの記述が簡素になることと、からである。この方式では、周期タスクは、最初に、イベントが発生しているかどうかを、入力モジュールのポーリングで判定し、発生時に、イベント処理を続けて行う。

ただし、特殊なエンジン制御では、OSの定義する周期以下で、タスクを起動したいことがある。この場合は、周期タスク機能を用いることはできない。かわりに、タイマポーリング関数をタスク内にもち、次の周期時間になるまで、ポーリングを行うことになる。

また、非特許文献であるOperating System Concepts - Fourth Edition - pp. 30-32 (Abraham Silberschatz 著: Addison-Wesley Publishing company 出版) には、以下のようなリアルタイム制御システムが開示されている。このリアルタイム制御システムは、制御の応答性を高めるために、制御対象物からの入力を、割込み信号として、演算機に通知し、割込みを発生させ、実行中に処理を中断させ、割込みハンドラから、制御対象物の処理を行うタスクを起動し、実行する。また、割込み処理実行中のタスクのリアルタイム性を保証するため、そのタスクの割込み中は、割込み禁止にしている。

さらに、特開2002-189606号公報の明細書には、周期的プログラムを一次停止し、再開するように構成することにより、再開時のオーバヘッド時間をなくすような周期タスク制御方法が開示されている。この他にも、特開2000-250764号公報の明細書には、複数あるタスクを機能毎に細分化して、独立タスクとすることにより、アプリケーションの切り替えオーバヘッド時間を少なくする処理実行装置が開示されている。

このように、同一タスクを繰り返し周期的に実行するエンジン制御などの場合は、周期タスクの起動周期が、最低でも1msec程度であり、又はタイマを使用して複数のタスクを実行する場合は、入力間隔が、1msec程度であった。そして、これらの時間は、CPU(50~60MHz動作)の処理速度に対して、比較的長い時間であるために、演算負荷はそれほど大きくなかった。しかし、

最近では、環境対応の要求などにより、より細かい制御が必要となってきた。例えば、電駆弁などを用いた自動車用エンジンでは、従来の周期よりも一桁以上短い周期で制御することが求められている。又は、複数のタスクを実行する場合でも、より木目細かいタイミング（数10 μ s）で制御することが求められている。

これらの条件を満たすためには、周期的にタスクを実行するような処理の場合は、その処理周期を短くする必要がある。従来の方法では、周期始めにあたる割込み処理に要するオーバヘッド時間は変えられないので、イベント処理の時間を短くなるような処理を行っている。しかし、これは、タイマ割込みの回数が多発するため、演算負荷が増大する。また、オーバヘッド時間よりも短い周期でタスクを実行することはできない。さらに、このような周期タスク方式では、各周期に発生するオーバヘッド時間が累積されることにより、イベントの検出が最大で1周期遅れる可能性もある。また、従来のタイマポーリング方式では、起動中のタスク以外のタスクを動作させることができないため、キャリブレーションなどの制御以外の処理を行えない。

さらに、複数の割込み処理を行う場合は、各々の処理が重複するような割込みを禁止しているので、それぞれ独立して、上記の処理が発生するため、各々の割込みには、オーバヘッド時間が数10～数100 μ s必要となる。このオーバヘッド時間は、非常に大きいので、上述した、要求は満たせない。また、複数の割込みを順に処理するので、個々の割込みのオーバヘッド時間が累積されて全体として処理の時間がかかり、後に実行されるタスクほど、その開始が遅れ、さらに、割込みが多い時は、すべての割込みに対してその要求を満たせないことも考えられる。

このように、複数の割込みが連続的、又は同時に発生した時に、個々の割込み時に発生するオーバヘッド時間を軽減するような構成に関わる技術は、これまでに開示されていない。

本発明は、このような問題に鑑みてなされたものであって、その目的とするところは、周期性のあるイベントが発生した場合又は複数のイベントが同時に発生

した場合において、周期タスクを的確に実施し、複数のオーバヘッド時間の累積による処理開始時間の遅れを防止することで、より細かな応答が可能な制御及び複数の指令に対して迅速かつ確実に応答可能な制御を行うリアルタイム制御システムを提供することにある。

発明の開示

前記目的を達成すべく、本発明に係るリアルタイム制御システムは、入力信号を受信し各種タスク処理に対応した割込み信号を出力するドライバ部と、該割込み信号に基づいてポーリングするポーリング部と、前記割込み信号に基づきタスク処理をするタスク処理部と、を備えたリアルタイム制御システムであって、前記ポーリング部は、前記タスク終了時に前記ポーリングに基づきタスク処理信号を出力し、前記タスク処理部は、前記タスク処理信号に基づいてタスク処理をする。

前記のごとく構成された本発明のリアルタイム制御システムは、ドライバ部で、割込み処理をし、割込み信号に基づいてタスク処理を行ない、それ以降の割込み信号は、ドライバ部で割込み処理を行わず、ポーリング部からのタスク処理信号に基づいてタスク処理される。すなわち、イベント処理をする際には、ドライバ部から、繰り返して割込み信号を受信して、タスクを起動しなくてもよく、タスク処理が連続時には、オーバヘッド時間は、初回の一回となる。

本発明に係るリアルタイム制御システムは、前記タスク処理部が、イベント処理を実行するイベント処理手段、及び該イベント処理の継続性を判定処理するタスク判定手段からなり、前記イベント処理手段は、前記判定結果が継続時に、前記イベント処理を継続実行する。

前記のごとく構成された本発明のリアルタイム制御システムは、入力信号に合わせたイベント処理手段を備えたとしても、タスク処理中に連続的又は同時に入力された複数の入力信号は、タスク判定処理を行なうので、一連の処理であるとみなすためオーバヘッド時間は、タスク処理始めの一回のみとなる。また、間隔が開いた入力の場合は、オーバヘッド時間とポーリング部の処理時間が発生する

が、ポーリング部の処理時間は、それほどの時間は要しないので、全体としての時間に影響はない。

本発明に係るリアルタイム制御システムは、前記イベント処理手段が、周期起動するイベント処理をし、タスク判定手段で判定されたイベント処理の継続又は終了の判定に従って、処理の実行を決定する。

前記のごとく構成された本発明のリアルタイム制御システムは、周期起動を行う際には、タスク内でイベント処理の継続又は終了の判定を行い、イベント処理手段のイベント処理の動作を決定するので、周期毎に、発生していたオーバーヘッド時間は、周期起動時に一回発生するのみである。

本発明に係るリアルタイム制御システムは、前記ポーリング部が、現時点の時刻をタイマからポーリングし、このポーリング結果に基づいて、前記イベント処理手段の開始時刻となるタスク処理信号を出力する。

前記のごとく構成された本発明のリアルタイム制御システムは、周期起動をするようなタスク処理の場合には、タスク処理信号が、タスク処理の開始時刻となる信号であってもよく、この開始時刻を決める信号は、本発明のシステムの外部のタイマをコールし、ドライバ部を経由して開始時刻決定をすることがないので、開始時刻をより早くタスク内の処理手段に送信することができる。

本発明に係るリアルタイム制御システムは、前記タスク判定手段が、前記タスク処理の継続回数、又は前記タスク処理中の割込み信号の有無に基づき、前記タスク処理の継続性を判定する。

前記のごとく構成された本発明のリアルタイム制御システムは、イベント処理の継続性を、その信号の割込み、イベント処理の内容、などのに合わせて決定できる。

本発明に係るリアルタイム制御システムは、前記ドライバ部、前記タスク処理部、及び前記ポーリング部と通信し、前記割込み信号の受信に伴い起動し、前記判定結果を格納するスケジューラを備える。

前記のごとく構成された本発明のリアルタイム制御システムは、ドライバ部、タスク、及びポーリング部が相互通信を行う時にスケジューラにデータを一時格

納することが可能となり、効率のよいタイミングにスケジューラからデータを伝達することができる。

本発明に係るリアルタイム制御システムは、前記スケジューラが、前記イベント処理手段に対応した周期を記録する周期テーブルを有し、前記イベント処理の完了時に、前記タスク判定手段は、前記周期テーブルから前記イベント処理の継続性を判定する。

前記のごとく構成された本発明のリアルタイム制御システムは、スケジューラに処理に対応した周期テーブルを設けているので、先に示したタイマからポーリングした時刻と、周期テーブルの周期から、ポーリング部で、対応したタスク処理の開始時刻を算出し、それらのデータをタスク処理部に送信することができる。また、周期テーブルから、イベント処理手段に周期性があるかの判定をすることが可能となり、この判定結果からも、イベント処理を継続するか決定できる。さらに、周期テーブルをスケジューラ内に設けているが、周期テーブルはスケジューラ外に設け、スケジューラが周期テーブルの周期を受信してもよい。

本発明に係るリアルタイム制御システムは、前記スケジューラが、前記タスク判定手段が起動中に、終了となる前記判定結果を格納するまでは、前記割込み信号が前記タスクに割込むことを禁止する。

前記のごとく構成された本発明のリアルタイム制御システムは、イベント処理途中に割込み信号の割込みを禁止しているので、タスク処理中に、割込みの影響を受けることが無く、確実に処理を実行できる。

本発明に係るリアルタイム制御システムは、前記ポーリング部が、前記イベント処理中に発生した前記割込み信号をポーリングし、該割込み信号に対応した前記イベント処理を実行するタスク処理信号を出力する。

前記のごとく構成された本発明のリアルタイム制御システムは、イベント処理中に、次のイベント処理の要求が発生したとしても、ポーリングしているので、このような割込みを禁止し再度ドライバ部に入力信号を送信する必要がない。よって、個々の割込み時に、オーバヘッド時間は発生しない。

本発明に係るリアルタイム制御システムは、前記タスク判定手段が、前記タス

ク処理信号の有無に基づいて、前記イベント処理の継続性を判定する。

前記のごとく構成された本発明のリアルタイム制御システムは、割込み信号がタスク処理信号であり、割込みが無い時は、イベント処理は終了することができる。

また、本発明に係るリアルタイム制御システムは、複数の前記タスク処理信号が、同時に検出された時に、前記タスク判定手段が、前記タスク処理信号の各々に優先順位を付けて、リードする。また、本発明に係るリアルタイム制御システムは、複数の前記タスク処理信号が、同時に検出された時は、前記タスク判定手段が、前記検出時に、前記イベント処理手段がイベント処理をしている処理に対応した前記タスク処理信号とは異なる信号を優先的にリードする。

前記のごとく構成された本発明のリアルタイム制御システムは、それぞれの入力信号に対応したイベント処理手段を実行し、複数の入力信号が入力されたときは、その処置順序に優先順位を付けたり、連続して同じ信号を処理しないような条件を設定してもよく、このように設定することで、割込みの要求に対応したイベント処理を的確に満すことができる。

図面の簡単な説明

図1は、本発明の実施形態に係るコントローラのハードウェアの構成を示すブロック図である。

図2は、本発明に係るリアルタイム制御システムの第一の実施形態を示すブロック図である。

図3は、タスクIDと、割込み因子に基づくイベントと、の対応を示すイベントテーブルである。

図4は、イベント毎の周期時間を示したテーブルである。

図5は、第一の実施形態に係るスケジューラの処理内容を示す制御フロー図である。

図6は、タスクのタスク処理内容を示す制御フロー図である。

図7は、タイマポーリング部の処理の詳細を示す制御フロー図である。

図8は、従来のイベント周期型タスクの処理を示すタイミングチャートである。

図 9 は、本実施形態に係るイベント周期型タスクの処理を示すタイミングチャートである。

図 10 は、本発明に係るリアルタイム制御システムの第二の実施形態を示すブロック図である。

図 11 は、第二の実施形態に係るタスクのタスク処理内容を示す制御フロー図である。

図 12 は、図 11 のポーリング処理内容を示す制御フロー図である。

図 13 は、入力 A の処理中に入力 B が入力された時の従来のタイミングチャートである。

図 14 は、入力 A の処理中に入力 B が入力された時の第二の実施形態に係るタイミングチャートである。

図 15 は、入力 A 処理後に入力 B が入力された時の第二の実施形態に係るタイミングチャートである。

発明を実施するための最良の形態

以下、本発明に係るリアルタイム制御システムの実施形態を図面に基づき詳細に説明する。

図 1 は、本発明の一実施形態に係るリアルタイム制御システム（コントローラ）10 のハードウェアの構成を示す制御ブロック図である。図 1 に示すように、コントローラ 10 は、CPU 11（演算装置）、メモリ 12 出力モジュール 13、及び入力モジュール 14、からなり、これらのハードウェアは、バスにより結合されている。

入力モジュール 14 は、入力信号である入力 A 及び入力 B を処理するハードウェアである。入力信号をレジスタに反映し、必要ならば、CPU 11 に割込みを入れる。そして、この入力モジュール 14 は、1 つ以上の入力信号を受けることが可能であり、信号が入力される毎に、CPU 11 に割込みを入れることができる。具体的には、入力モジュール 14 は、入力信号を受けたポートを識別するためのフラグを持ち、フラグを CPU 11 に割込み要因として与えることができる。

。

また、CPU 11は、演算機をもち、入力モジュール14からの信号を割込み要因として受取り、この割込み要因を受けて、メモリ12から命令となる信号をリードし、実行する。この他にも、CPU 11は、タイマ15、及び内蔵メモリを有している。タイマ15を内蔵しないCPUを用いる場合は、外付けのタイマを用いてもよい。このタイマ15は、経過時刻をCPU 11に示すカウンタである。本実施形態では、タイマ15はアップカウンタとするが、タイマ15がダウンカウンタの場合でも、実施可能である。

また、メモリ12は、不揮発性メモリ、及び／又は、揮発性メモリを有し、CPU 11からの入力信号を受けて、CPU 11が行う演算の命令となる信号をCPU 11に送る。不揮発性メモリとしては、EEPROM又はフラッシュメモリなどがあり、不揮発性メモリは、プログラムや、データの初期値が記録される。一方、揮発性メモリとしては、SRAMS又はSDRAMなどがあり、揮発性メモリは、データ、及び／又は、スタックが記録される。実施形態のリアルタイム制御装置を実施するオペレーティングシステム（以下OSと云う）も、メモリ12に記録される。

さらに、出力モジュール13は、CPU 11からの演算結果に基づいて、出力信号を処理するハードウェアである。出力モジュール13は、1つ以上の出力ポートを持つことができる。ちなみに、本実施形態では、出力モジュール13は、なくてもよい。

図2は、第一の実施形態のリアルタイム制御システムのソフトウェアを示すブロック図である。本実施形態のソフトウェアは、スケジューラ1、タイマポーリング部2、周期テーブル3、ドライバ部4、タスク処理部（以下タスクと云う）5及びCPU 11にあるタイマ15により構成される。

スケジューラ1、タイマポーリング部2、ドライバ部4、及びタスク5は、OSに含まれるが、ミドルウェアとしてOSの外に存在してもよい。また、タスク5は、アプリケーションの要求にあわせて、2つ以上存在してもよい。

まず、ドライバ部4は、ドライバ部4内にある割込みハンドラ、及び入力モジ

ジュール14のレジスタを初期設定、及び／又は、リード及びライトするデバイスドライバからなる。すなわち、ドライバ部4は、リセット直後に、入力ジュール14や出力ジュール13のレジスタを初期化する。また、先に示した入力ジュール14の割込み要因となる入力信号の受信割込みをする必要がある場合には、ドライバ部4は、入力ジュール14から入力A又は入力Bなどからなる入力信号の割込みを許可するように設定される。そして、ドライバ部4の割込みハンドラは、割込みが発生した直後に、入力ジュール14から入力信号をリードし、スケジューラ1に、割込み信号を送信する。

また、スケジューラ1は、ドライバ部4の割込みハンドラが、入力ジュール14からリードした入力A又は入力Bの割込み信号を受信することにより、起動する。

図3は、後述するタスク5内に存在する複数のイベント処理手段51を識別するタスクIDと、ドライバ部4で入力された入力A又は入力Bの割込み要因に基づくイベント（事象）と、の対応関係を表したイベントテーブルであり、スケジューラ1は、このイベントテーブルを有している。そして、スケジューラ1は、入力A又は入力Bのイベントに従って、イベントテーブルから、タスクIDを検索し、検索したタスクIDをタスク5に送信する。ちなみに、スケジューラ1は、入力Aの割込みをタスクID=0、入力Bの割込みをタスクID=1と識別し、タスクIDをタスク5に送信し、タスク5を起動させる。また、スケジューラ1は、2タスク分のデータを保存しているが、3以上のタスクでも可能である。さらに、スケジューラ1は、タスクID送信後に、タスク5からの信号である後述する戻り値を受信し、周期テーブル3からタスク周期時間を読み取り、そしてタイマポーリング部2に引数を与え、その結果、タスク5の次のタスク処理開始時刻であるタスク処理信号をタスク5に送る。

また、タスク5は、アプリケーション処理を実行するプログラムであり、タスクIDに対応した複数のイベント処理を行うイベント処理手段51と、イベント処理の継続又は終了を判定するタスク判定手段52と、を有する。そして、タスク5のイベント処理関数を有したイベント処理手段51を実行するために、スケ

ジューラ 1 のイベントテーブルから検索されたタスク ID は、エントリアドレスにジャンプすることで、タスク 5 を起動し、先に示したタスク判定手段の判定結果を、スケジューラ 1 にリターンする。

この他にも、実施形態に係るタスク 5 のイベント処理手段 5 1 は、イベントタスクと、イベント起動型周期タスクと、を有してもよく、スケジューラ 1 は、この 2 種類のイベント処理手段 5 1 をサポートする。他の実施形態では、周期タスクをサポートしてもよい。周期タスクをサポートする場合は、周期タスクのタイマの割込みが発生する。これが、イベント起動型周期タスク実行中に発生すると、タスクに待ち時間が発生し、周期時間をオーバーすることがある。これを回避するために、スケジューラ 1 は、イベント起動型周期タスクを起動するときに、割込み禁止とし、このイベント起動型周期タスクを終了するときに、割込みを許可とするようになっている。

また、周期テーブル 3 は、図 4 に示す、イベント毎の周期時間を示したテーブルを有している。この周期テーブル 3 は、スケジューラ 1 からのタスク ID を受信して、タスク ID に対応した各イベントの周期時間を検索し、スケジューラ 1 に送信するプログラムである。ちなみに、この表では、タスク ID = 0 の時は、イベントは、イベント起動型周期タスクであり、タスク = 1 の時は、周期性を持たないイベントタスクであることを示している。また、2 タスク分のデータを保存しているが、3 以上のタスクでも可能である。

タイマポーリング部 2 は、目標時間までタイマ 1 5 をポーリングし、目標時刻になるとスケジューラ 1 に、タスク処理の起動時刻であるタスク処理信号を、リターンするプログラムである。スケジューラ 1 は、タスク 5 の次の起動時刻までに、タイマポーリング部 2 をコールする。スケジューラ 1 は、タスク 5 の起動型周期タスクであるイベント処理手段 5 1 の周期を、引数としてタイマポーリング部 2 に与える。本実施形態では、入力 A 又は入力 B の入力によりスケジューラ 1 を起動するため、入力 A 及び入力 B の割込みを許可しておく。

図 5 は、第一の実施形態に係るリアルタイム制御システムのスケジューラ 1 の処理内容を示す制御フロー図である。スケジューラ 1 は、ドライバ部 4 から割込

みを受けると実行開始する。まず、ステップ501では、先に示した割込み信号をドライバ部4からリードする。この割込み信号は、どのような割込みであるかを識別するためのフラグであり、ここでは、「入力A」又は「入力B」に対応したフラグである。

ステップ502で、図3に示したイベントテーブルを用いて、割込み信号により起動させるタスク5を検索する。起動するタスク5がない場合は、スケジューラを終了し、起動するタスク5のタスクIDがある場合は、ステップ503に進み、タスク5を起動させる。すなわち、入力Aが割込んだ場合は、タスクID=0を与え、入力Bが割込んだ場合は、タスクID=1を与え、タスク5を起動させる。このタスク5の起動は、タスク5のエントリアドレスにジャンプすることにより実現する。また、コンテキストスイッチが必要な場合は、CPU11のレジスタを全て保存し、レジスタをクリアする。ステップ503でタスク5が起動し、イベント処理手段51の処理が終了すると、ステップ504に進む。ここに示すステップ503の詳細は、後述する図6で説明する。

ステップ504では、スケジューラ1は、タスク5のタスク判定手段52から、戻り値として「継続」又は「終了」のどちらかの信号を格納する。戻り値が「終了」の場合は、スケジューラ1は、終了され、戻り値が、「継続」の場合は、ステップ505に進む。

ステップ505は、先に示したタスクIDを用いて、図4に示す周期テーブル3を検索し、タスク5の周期が0かどうか（周期をもつかどうか）を判定する。例えば、タスクID=0（イベント：入力A）の場合は、周期は10usecとなり、タスクID=1（イベント：入力B）の場合は、周期は0usecとなる。そして、周期が0（タスクID=1）の場合は、スケジューラ1を終了させ、それ以外（タスクID=2）の場合は、タスク5は、イベント起動型周期タスクであると判断し、ステップ506に進む。

ステップ506では、タイマポーリング部2を実行し、次のタスク起動時刻まで待つ。タイマポーリング部2には、スケジューラ1で検索した周期を、引数として与える。このタイマポーリング部2の実行の詳細は、後述する図7で説明す

る。その後、ステップ504にあるタスク5からの戻り値が「終了」となるまで、一連のステップ503～ステップ506を繰り返し実行する。

図6は、図5に示すステップ503のタスクの処理を示す制御フロー図である。まず、ステップ601では、タスク5のタスク判定手段52が「継続」するか「終了」するかどうかの判定処理を行う。判定処理は、例えば「入力Aの割込みが発生しているかどうか」、「イベント発生後、タスク5が起動しイベント処理手段51を実行した回数が、規定の回数に達したかどうか」などの判定基準を設け、その判定基準に従って、判定処理を行い、ステップ602に進む。ステップ602では、ステップ601のタスク判定手段52の判定処理に基づいて、判定結果を判断する。判定結果が「終了」であれば、ステップ603に進み、「終了」を戻り値としてスケジューラ1に返し、タスク5を終了させる。判定結果が「継続」であれば、ステップ604に進み、イベント処理を行う。イベント処理では、CPU11が、入力A又は入力Bに対応したイベント処理手段51を用いて実行する。出力があれば、出力モジュール13から、信号を出力する。その後、ステップ605に進み、「継続」を戻り値としてスケジューラ1に返し、終了する。

図7は、タイマポーリング部2の処理を示す制御フロー図である。まず、ステップ701で、現在時刻 T_c （以下 T_c ）と前回ポーリングが終了したポーリング終了時刻 T_o （以下 T_o ）とをリードする。 T_c は、現時点でのタイマ15の値である。また、 T_o は、タイマポーリング部2が前回の実行時に保存した時の時刻である。イベント発生直後の場合は、タイマポーリング部2をはじめて実行するので、ステップ701の実行時には、 T_o は不定となる。そこで、この問題を回避するために、図5のステップ501とステップ502の間で、スケジューラ1がタイマ15をリードし、タイマポーリング部2の T_o として記録しておくといふ。

次に、ステップ702で、次に処理される予定であるタスク5のイベント処理手段51の起動時刻 T_n （以下 T_n ）を求める。 T_n は、" T_o +周期- T_c "により求まる。 T_o から T_c の間でタイマ15がオーバーフローした場合、すなわち、

T_o よりも T_c が小さい場合、その分を補正して T_n を決定する。

その後、ステップ703で、タイマ15から T_c を再度リードする。ステップ704で、 T_c が T_n よりも大きいかどうか判定する。 T_c と T_n の間で、タイマ15のオーバフローが発生する場合、すなわち T_c よりも T_n が小さい場合、タイマ15がオーバフローになるまでは、 T_n と T_c の比較は行わない。ステップ704の比較の結果、真になるまで、ステップ703及びステップ704を繰り返す。ステップ704が真になると、ステップ705に進み、 T_c を T_o に保存する。

図8及び図9は、イベント処理手段51であるイベント周期型タスクの処理を示すタイミングチャートであり、図8は、従来のタイミングチャートであり、図9は、本実施形態に係るタイミングチャートである。

図8に示すように、従来の方式では、周期タスクは、最初に、イベントが発生しているかどうかを、入力モジュールのポーリングで判定し、発生時に、イベント処理を続けて行う。入力Aの割込み発生後、割込み処理にオーバヘッド時間 T_i （以下 T_i ）かかる。その後、タスク処理中で、判定処理に処理判定時間 T_j （以下 T_j ）、イベント処理にイベント処理時間 T_t （以下 T_t ）かかる。これを繰り返すので、本方式の最小のタスク周期は、 $T_i + T_j + T_t$ となる。

一方、図9に示すように、本実施形態では、入力Aが入力されるとドライブ部の割込みハンドラが起動される。そのとき、割込み処理時間は、 T_i かかる。その後、スケジューラ1がタスク5を起動し、判定処理を行う。図8に示すように T_j と同様に判定処理時間は T_j かかる。判定処理が「継続」の場合、イベント処理を行い、イベント処理時間 T_t （以下 T_t ）の間で実行される。イベント処理終了後、次の周期まで時間があれば、スケジューラ1が、タイマポーリング部2を起動し、ポーリング時間 T_p （以下 T_p ）だけ待つ。以降は、判定処理が「終了」と判断するまで、 T_j, T_t, T_p を繰り返す。従って、タスク5の周期は、 $T_j + T_t + T_p$ となり、この周期最小値は $T_p = 0$ ときなので、 $T_j + T_t$ となる。ここで、 T_i は数10 μs 程度であるのに対して T_p は1 μs 以下であるので、タスクの周期は大幅に短縮することができる。

次に、図10は、本発明のリアルタイム制御システムの第二の実施形態を示す制御ブロック図である。本実施形態のリアルタイム制御システムのソフトウェアは、ドライバ部4'、ポーリング部2'、及びタスク処理手段（以下タスクと云う）5'より構成される。タスク5'は、アプリケーションの要求にあわせて、2つ以上存在することができる。

ドライバ部4'は、割込みハンドラ、及び、入力モジュール14のレジスタを、初期設定、若しくはリード、又はライトするデバイスドライバである。ドライバ部4'は、リセット直後に、入力モジュール14や出力モジュール13のレジスタを初期化する。受信割込みが必要な場合は、入力モジュール14からの割込みを許可するように設定する。本実施形態では、入力信号である入力Aと入力Bの割込みによりタスク5'を起動するため、ドライバ部4'は入力A、入力Bの割込みを予め許可している。割込みハンドラは、割込みが発生した直後に実行される。ドライバ部4'は、入力モジュール14から入力信号をリードし、割込み信号を出力し、タスク5'を呼び出す処理を行う。本実施形態では、入力が入力Aと入力Bとの2つだが、3個以上の入力があってもよい。

また、タスク5'は、アプリケーション処理を実行するプログラムである。タスク5'のエントリアドレスは、ドライバ部4'が記憶しており、タスク5'は、一つ以上の関数から構成されるが、本実施形態では、少なくとも、タスク判定手段52'と、入力Aの処理を行う入力A関数を有したイベント処理手段51Aと、入力Bの処理を行う入力処理関数Bを有したイベント処理手段51Bと、を有する。

また、ポーリング部2'は、ドライバ部4'の入力A又は入力Bに対応した割込み信号をポーリングし、タスク5'にタスク処理信号を出力するプログラムである。タスク5'のイベント処理手段51A、イベント処理手段51Bの実行が終了したときに、すぐにタスク5'は終了されず、割込み因子の有無を確認するために、ポーリング部2'は、タスク判定手段52'からコールされるタスク処理信号を出力する。

図11は、第二の実施形態に係るリアルタイム制御システムのタスク5'の処

理内容を示す制御フロー図である。まず、ステップ1101では、タスク5'は、ドライバ部4'より起動された直後、ドライバ部4'より引数として渡された割込み信号をリードする。そしてステップ1102に進み、ステップ1102では、割込み要因である割込み信号を”入力A”又は”入力B”のいずれの割込みであるかタスク判定手段52'で判定する。ここで、”入力A”ならば、ステップ1103に進み、入力Aに対応したイベント処理手段51Aの入力A関数を実行する。入力A関数は、入力Aの割込みが発生した時に行う関数である。例えば、入力Aが、エンジンコントローラユニットのクランク信号であれば、イベント処理手段51Aでは、点火タイミングの計算などを行う。ステップ1102において、割込み信号が、”入力B”であれば、ステップ1104に進み入力B関数を実行する。もし、ステップ1102において、割込み信号が”なし”であれば、タスク5'を終了する。次に、ステップ1105において、タスクのタスク判定手段52'はポーリング部2'をコールし、タスク処理信号があれば受信する。その後、再び、タスク処理信号を用いてステップ1102の判定を行い、ステップ1102の判定が、”なし”になるまで、先に示した一連の作業を繰り返す。

ステップ1102の判定は、同時に複数の信号が存在する時の選択方式により、2種類に分類される。第一に、信号を交互に選択する方式である。これは、前回検出した信号を、メモリに記憶しておき、これ以外の信号を選択する方式である。第二に、割込み要因を定まった順序（優先順位を付けた順序）で選択する方式である。第一の方式の場合、入力A又は入力Bを交互に選択するので、公平に処理することができる。例えば、入力A、入力Bの処理に優先順位が存在しないようなシステムに適している。第二の方式の場合は、入力Aを必ず選択し、入力Aが無い場合のみ入力Bを選択する。あるいは、入力処理を優先的に実行してもよい。例えば、入力Aの処理の優先順位が、入力Bの順位よりも高いようなシステムに適している。本実施例では、第一の方式、第二の方式を示したが、これら以外の順序で、入力A、入力Bを選択してもよい。

さらにステップ1102の判定は、同一信号が、再度検出された時の判定方式

によっても、二種類に分類される。第一に、再度検出された信号に対応するイベント処理手段51A、又はイベント処理手段51Bを実行する方式である。第二に、再度検出された信号は無視し、未検出の信号を選択する方式である。第一の方式のメリットは、連続して入力がある場合、割込みハンドラのオーバーヘッド時間なしに、複数回の入力関数を実行できることである。

そしてこのような第一の方式は、入力Aが周期的な入力であり、かつ、その周期的が割込みハンドラのオーバーヘッドに比べて、極めて短い場合に適している。第二の方式のメリットは、定期的に割込み許可状態にできることである。本実施形態では、タスク5'は、リアルタイム性を確保する（優先順位が高い）ために割込み禁止で動作しているが、タスク5'が動作し続けると、入力A又は入力B以外の割込みが入った時に、処理できない。そこで、第二の方式を用いることで、入力A、入力Bの処理を一回行ったあとに、タスク5'を一旦終了させることで、他の割込みが入れるようにする。

図12は、図11のステップ1105のポーリング処理内容を示す制御フロー図である。まず、ステップ1201で、ドライバ部4'から割込み信号をリードする。割込み信号は、入力モジュール14、出力モジュール13いずれかのレジスタに、一括、又は分散して記録される。これらのうち必要な割込み信号を、リードする。本実施形態では、タスク5'は入力Aと入力Bの割込み処理を行うので、入力A及び入力Bの割込み信号をリードする。そしてステップ1202において、割込み信号を、タスク5'に戻り値として返す。入力の数が、32個以内であれば、入力ごとの割込み要因をビットマップに置き換えて4バイトコードにし、関数の戻り値として返すこともできる。もし、32個以上の入力があれば、割込み要因を保存する構造体を生成し、そのポインタをリターンしてもよい。

図13～図15は、イベント処理手段51A及びイベント処理手段51Bの処理を示すタイミングチャートであり、図13は、入力Aの処理中に入力Bが入力された時の従来のタイミングチャートであり、図14は、入力Aの処理中に入力Bが入力された時の本実施形態に係るタイミングチャートであり、図15は、入力A処理後に入力Bが入力された時の本実施形態に係るタイミングチャートであ

る。

図13に示すように、従来では、入力Aの割込みが発生してから、割込みハンドラの処理を行う時間であるオーバヘッド時間 T_i を要する。その後、入力Aの処理を行うイベント処理手段51Aを処理時間 T_t の間、実行する。イベント処理手段51Aの処理中に、入力Bの割込みが発生しても、割込み禁止されているので、イベント処理手段51Aの処理を実行する。その後、イベント処理手段51Aの処理が終了すると、割込み禁止が解除されて、入力Bの割込みが演算機により検出される。そして、再び割込みハンドラがコールされ、オーバヘッド時間 T_i 後に、イベント処理手段51Bを実行する。

図14に示すように、本実施形態では、まず、入力Aの信号が入力されると、従来と同様に、ドライバ部4'の割込みハンドラが、オーバヘッド時間 T_i の間、割込み処理を行う。しかし、イベント処理手段51Aの処理中に、入力Bの割込みが発生すると、この信号が、ポーリング部2'に格納され、関数終了後、ポーリング時間 T_p だけポーリングを行い、イベント処理手段51Bを実行する。ここでも、第一の実施形態と同様に、 T_i は数10 μs 程度であるのに対して T_p は1 μs 以下であるので、タスクの周期は大幅に短縮することができる。

図15に示すように、入力Aと入力Bの入力間隔が大きい場合には、各々の入力には、オーバヘッド時間 T_i に加え、ポーリング時間 T_p が必要となるので、ポーリング時間 T_p だけ無駄な処理を行うことになるが、この時間も前述したように、極めて短い時間であるので、性能に悪影響を及ぼすことはない。

以上、本発明の一実施形態について詳述したが、本発明は、前記の実施形態に限定されるものではなく、特許請求の範囲に記載された本発明の精神を逸脱しない範囲で、種々の設計変更を行うことができるものである。

例えば、ドライバ部、タスク、スケジューラ、ポーリング部などは、その機能に合わせたプログラム内容であって、これらの構成が統合及びさらに分割されても、その機能に変わりはない。

また、第一の実施形態と第二の実施形態とを別個に説明したが、これらの実施

形態を複合させるような実施形態であってもよい。特に、割込みされたタスク処理信号として、入力信号、タイマの信号などを用いたが、これらの信号に限られるものではない。

また、これらのリアルタイム制御システムは、例えば、自動車のコントローラ（エンジンプリクラシュ・シート、シートベルト・プリテンショナー、エアバッグ、クラッチ、アンチブロックシステム、電駆弁エンジン等のコントローラ）や、一般の産業機器のコントローラなどに適用可能であり、割込み信号が入力されるようなシステムであるならば、特に制限されるものではない。

産業上の利用の可能性

以上の説明から理解できるように、本発明のリアルタイム制御システムは、周期性のあるイベントが発生した場合又は複数のイベントが同時に発生した場合において、周期タスクを的確に実施し、複数のオーバーヘッド時間の累積することによる処理開始時間の遅れを防止することで、より細かな応答が可能な制御及び複数の指令に対して迅速かつ確実に応答可能な制御を行うことができる。

請 求 の 範 囲

1. 入力信号を受信し各種タスク処理に対応した割込み信号を出力するドライバ部と、該割込み信号に基づいてポーリングするポーリング部と、前記割込み信号に基づきタスク処理をするタスク処理部と、を備えたリアルタイム制御システムであって、

前記ポーリング部は、前記タスク終了時に前記ポーリングに基づきタスク処理信号を出力し、前記タスク処理部は、タスク処理信号に基づいて前記タスク処理をすることを特徴とするリアルタイム制御システム。

2. 前記タスク処理部は、イベント処理を実行するイベント処理手段、及び該イベント処理の継続性を判定処理するタスク判定手段からなり、前記イベント処理手段は、前記判定結果が継続時に、前記イベント処理を継続実行することを特徴とする請求項1に記載のリアルタイム制御システム。

3. 前記イベント処理手段は、周期起動するイベント処理をすることを特徴とする請求項1又は2に記載のリアルタイム制御システム。

4. 前記ポーリング部は、タイマをポーリングし、前記タスク処理の開始時刻となる前記タスク処理信号を出力することを特徴とする請求項1～3のいずれか一項に記載のリアルタイム制御システム。

5. 前記タスク判定手段は、前記タスク処理の継続回数、又は前記タスク処理中の割込み信号の有無に基づき、前記タスク処理の継続性を判定することを特徴とする請求項2～4のいずれか一項に記載のリアルタイム制御システム。

6. 前記ドライバ部、前記タスク処理部、及び前記ポーリング部と通信し、前記割込み信号の受信に伴い起動し、前記判定結果を格納するスケジューラを備えることを特徴とする請求項2～5のいずれか一項に記載のリアルタイム制御システム。

7. 前記スケジューラは、前記イベント処理手段に対応した周期を記録する周期テーブルを有し、前記タスク判定手段は、前記イベント処理の完了時に、前記周期テーブルの信号に基づき前記イベント処理の継続性を判定することを特徴とする請求項6に記載のリアルタイム制御システム。

8. 前記スケジューラは、前記タスク判定手段が起動中に、終了となる前記判定結果を格納するまでは、前記割込み信号が前記タスクに割込むことを禁止することを特徴とする請求項6又は7に記載のリアルタイム制御システム。

9. 前記ポーリング部は、前記タスク処理中に発生した前記割込み信号をポーリングし、該割込み信号に対応した前記イベント処理を実行する前記タスク処理信号を出力することを特徴とする請求項1又は2に記載のリアルタイム制御システム。

10. 前記タスク判定手段は、前記タスク処理信号の有無に基づいて、前記イベント処理の継続性を判定することを特徴とする請求項9に記載のリアルタイム制御システム。

11. 複数の前記タスク処理信号が、同時に検出された時は、前記タスク判定手段は、前記タスク処理信号の各々に優先順位を付けて、リードすることを特徴とする請求項2又は10に記載のリアルタイム制御システム。

12. 複数の前記タスク処理信号が、同時に検出された時は、前記タスク判定手段は、前記検出時に、前記イベント処理手段がイベント処理をしている処理に対応した前記タスク処理信号とは異なる信号を優先的にリードすることを特徴とする請求項2又は10に記載のリアルタイム制御システム。

図 1

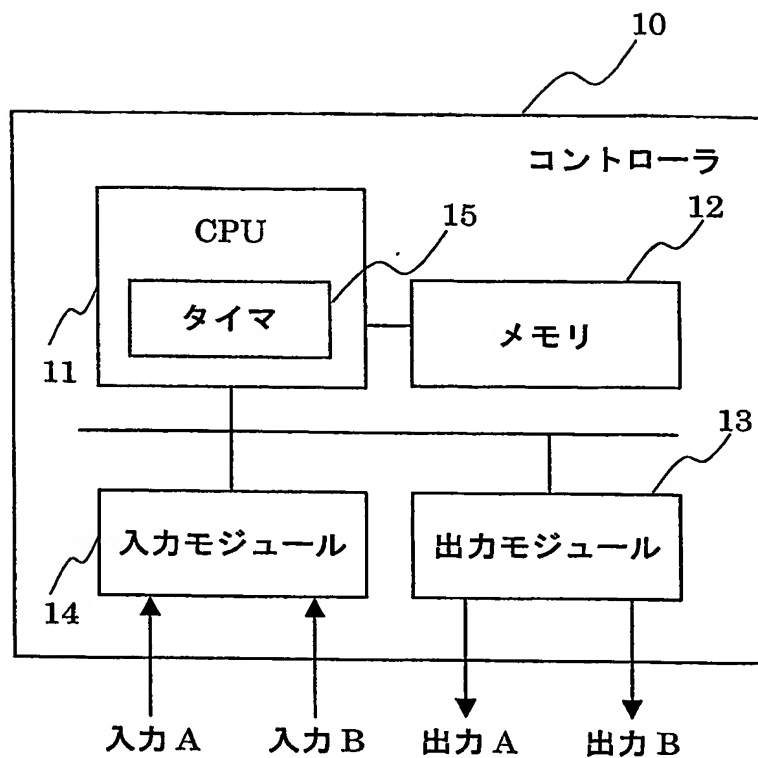


図 2

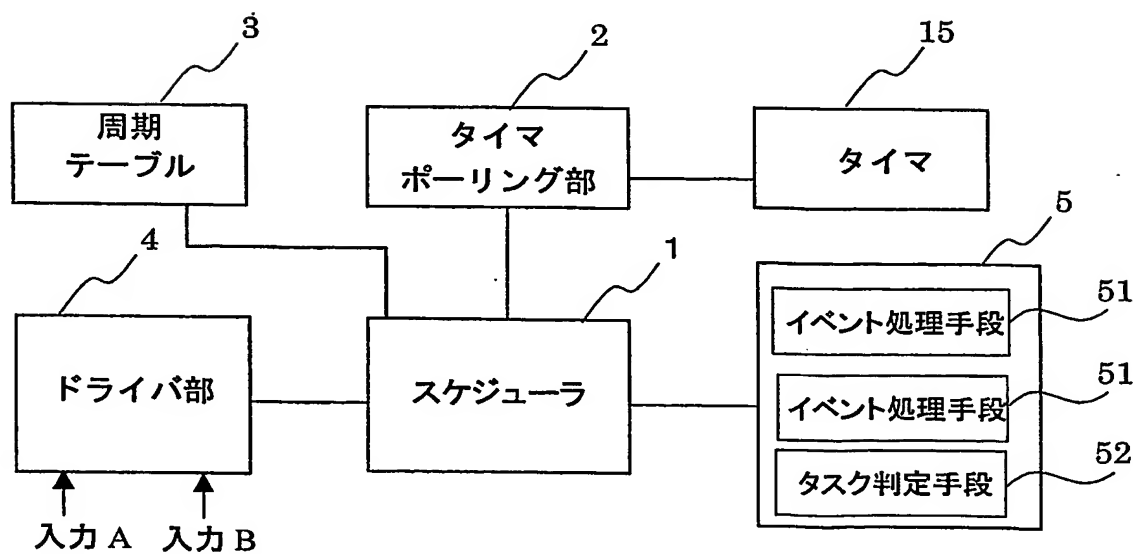


図 3

| タスクID | イベント |
|-------|------|
| 0 | 入力A |
| 1 | 入力B |

図 4

| タスクID | 周期 |
|-------|----|
| 0 | 10 |
| 1 | 0 |

図 5

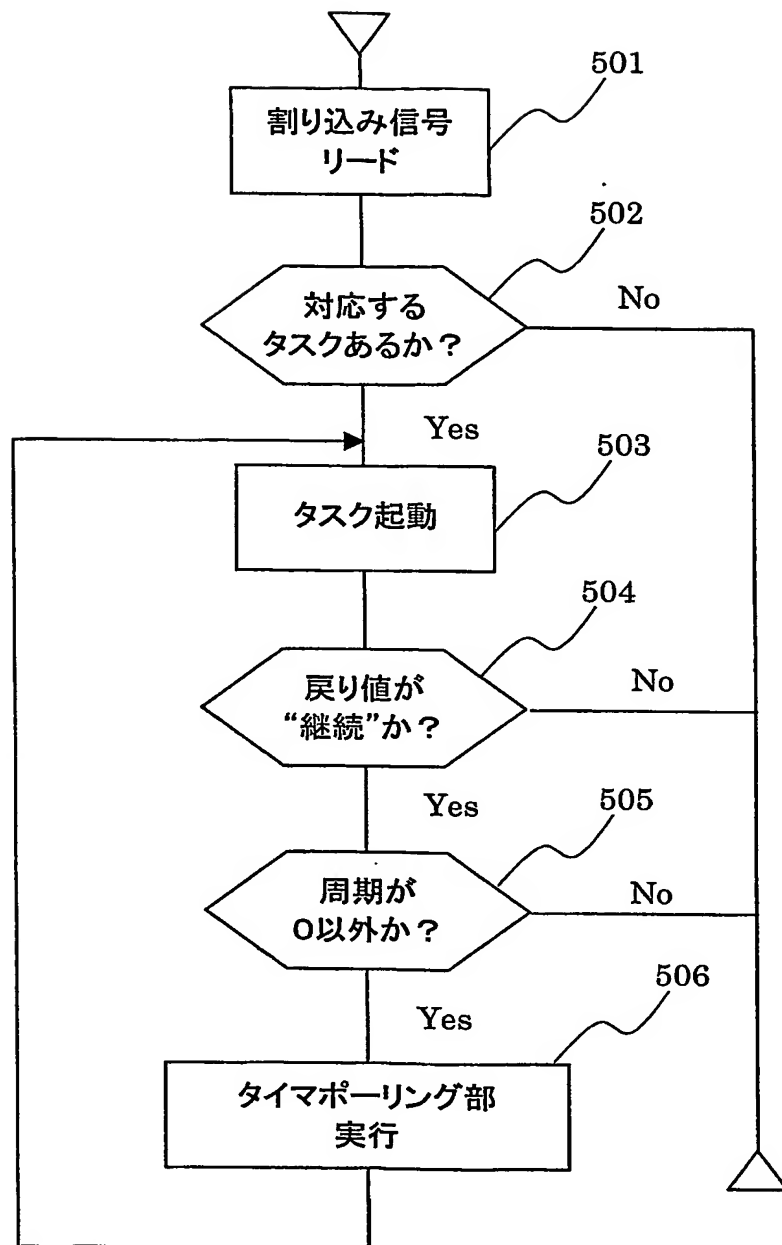


図 6

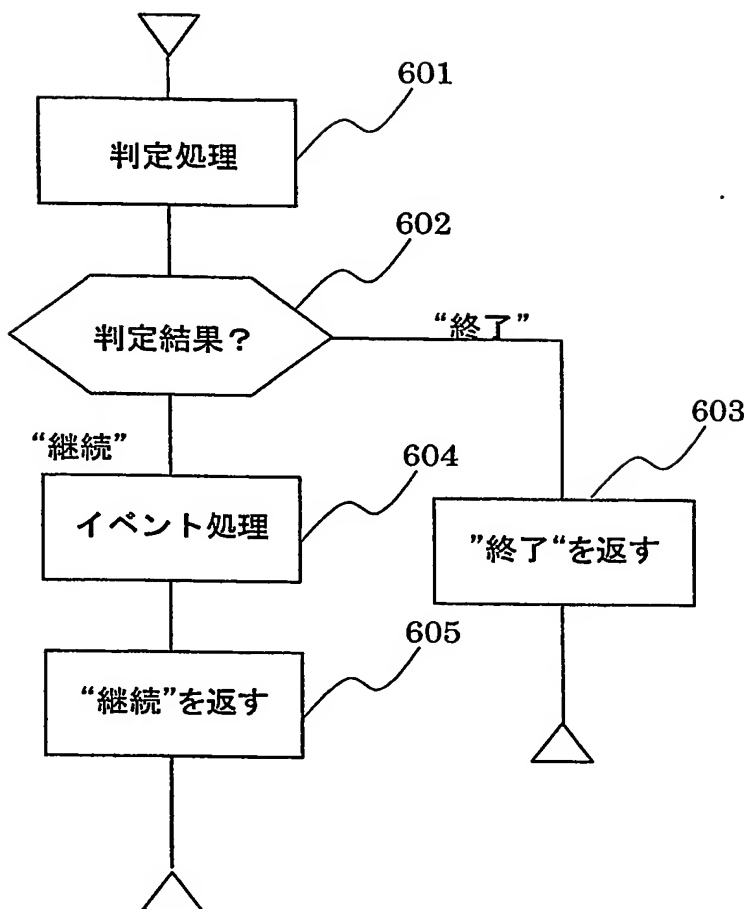


図 7

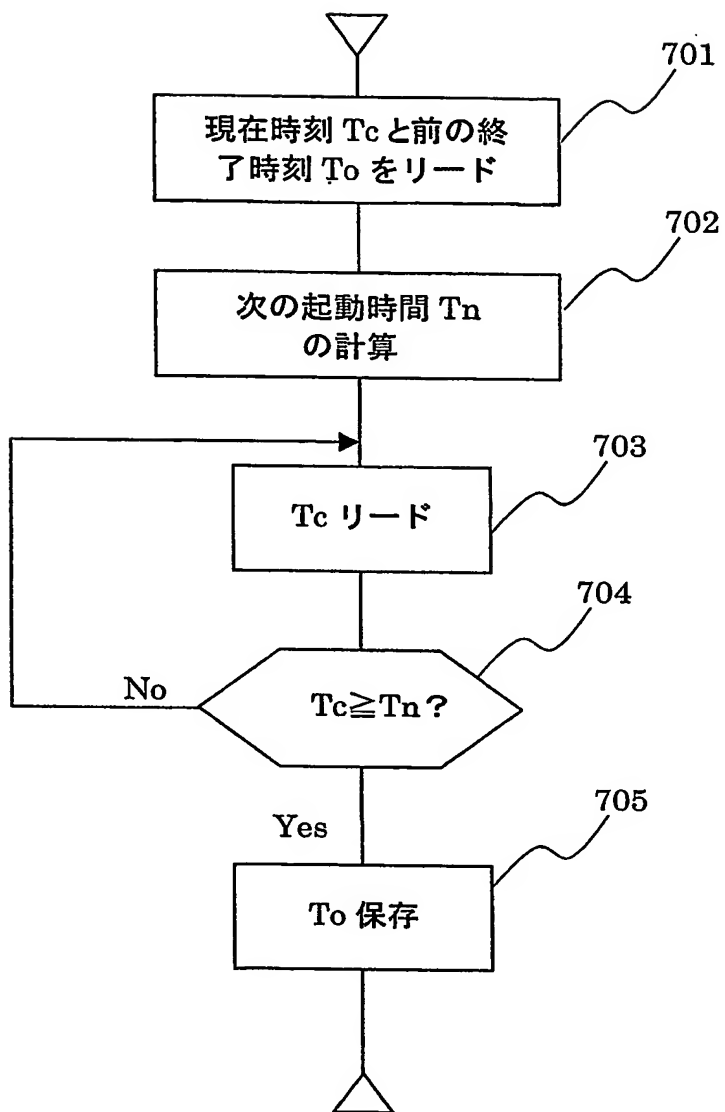


図 8

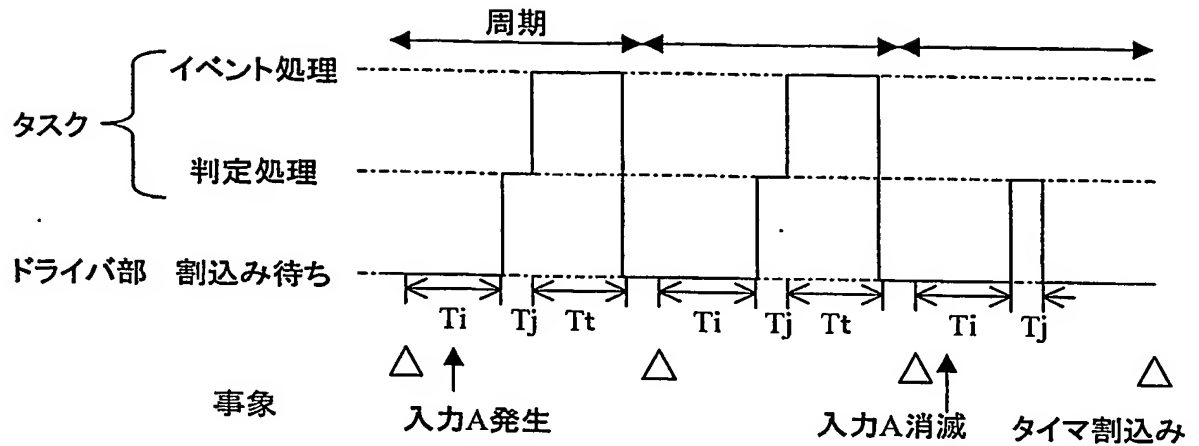


図 9

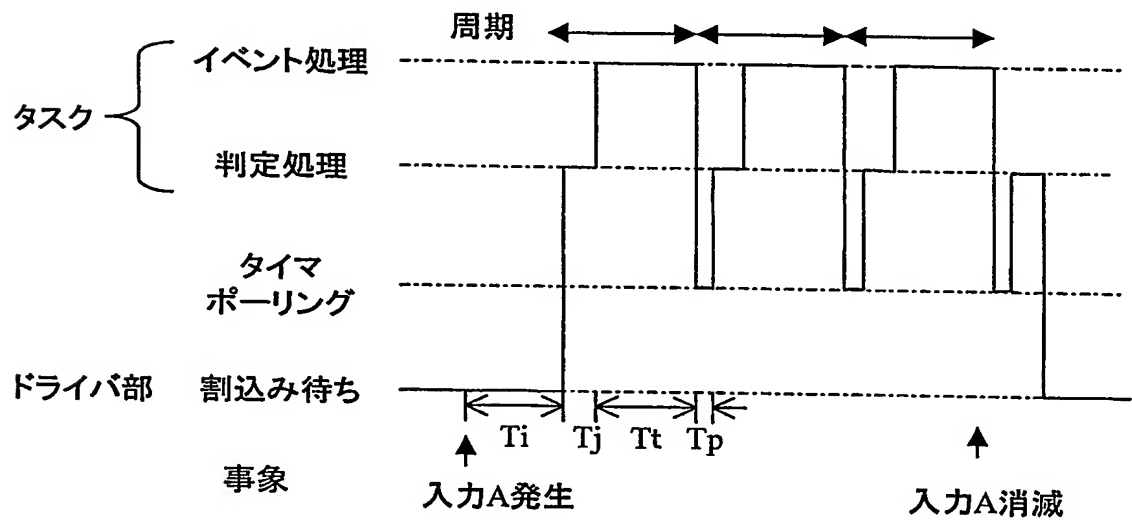


図 1 0

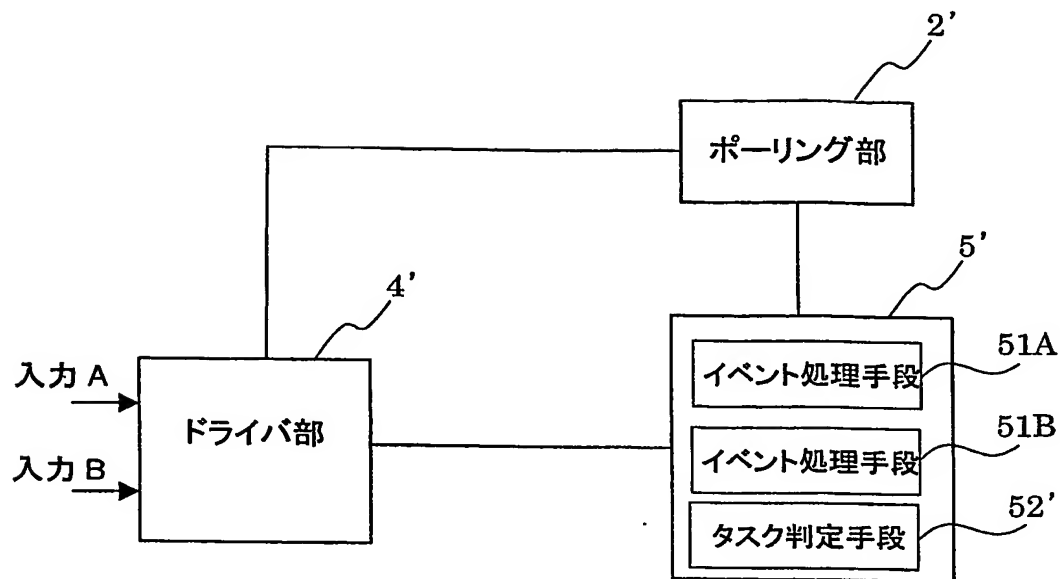


図 1 1

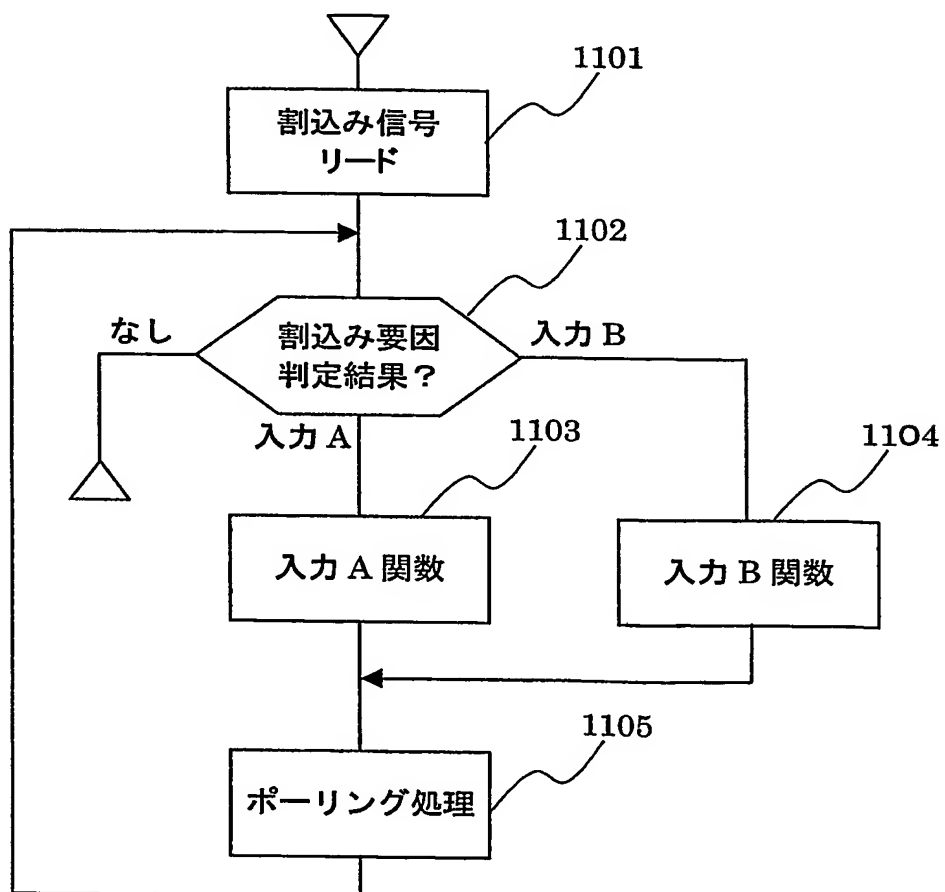


図 1 2

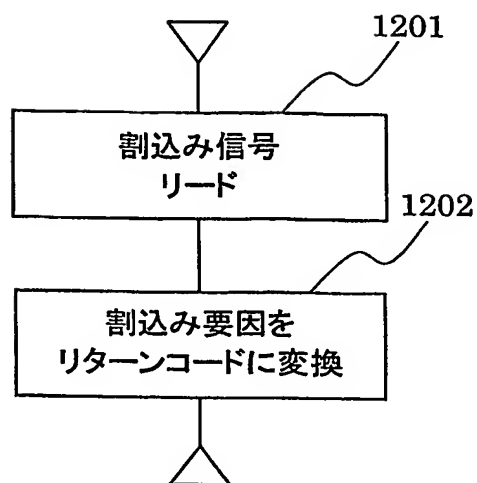


図 1 3

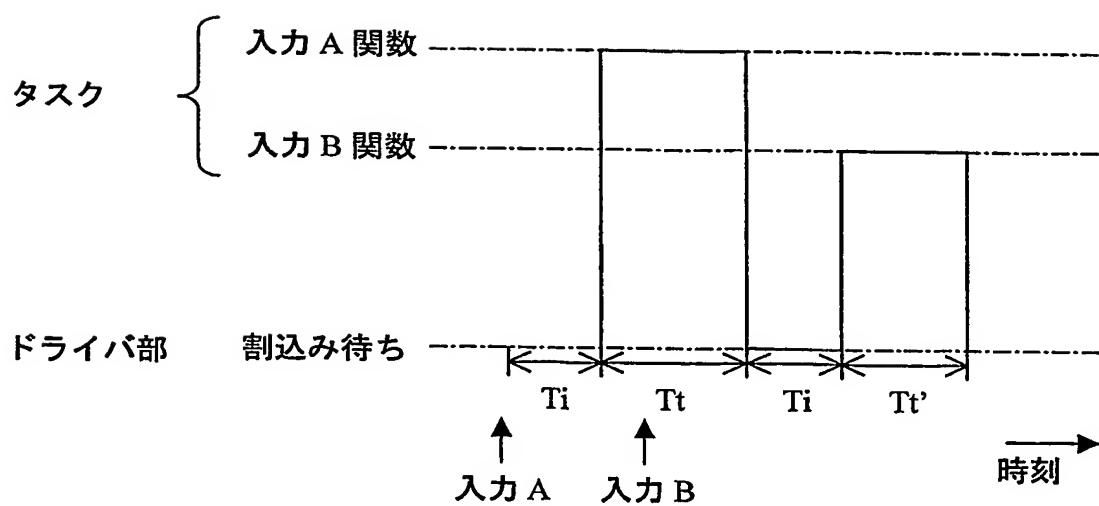


図 1 4

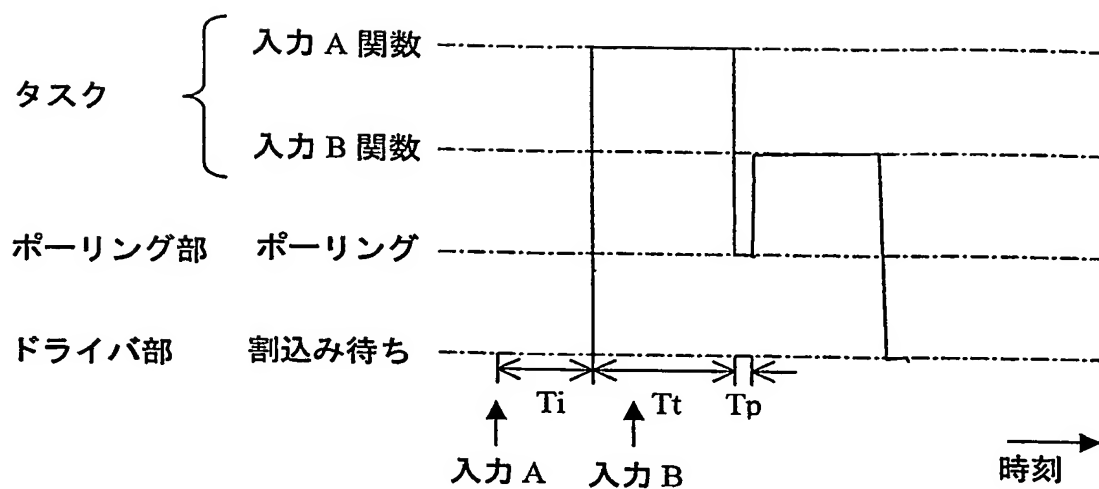
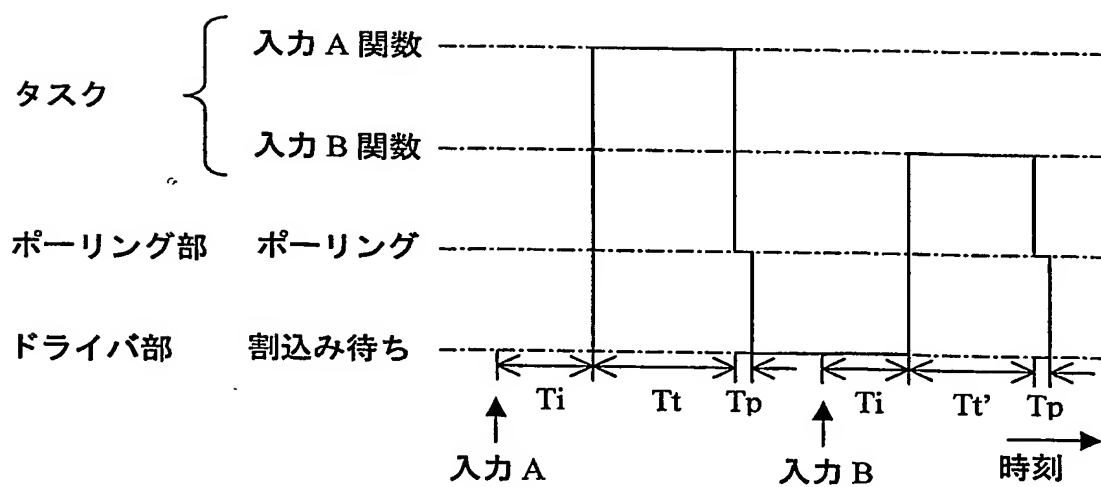


図 1 5



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/09877

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl⁷ G06F9/48, F02D45/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁷ G06F9/46-9/54, F02D43/00-45/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

| | | | |
|---------------------------|-----------|----------------------------|-----------|
| Jitsuyo Shinan Koho | 1922-1996 | Toroku Jitsuyo Shinan Koho | 1994-2003 |
| Kokai Jitsuyo Shinan Koho | 1971-2003 | Jitsuyo Shinan Toroku Koho | 1996-2003 |

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|-----------|---|-----------------------|
| X | JP 2003-15885 A (Canon Inc.), | 1, 2, 5-10 |
| Y | 17 January, 2003 (17.01.03), Par. No. [0007] (Family: none) | 3, 4, 11, 12 |
| Y | Takanori YOKOYAMA et al., "Kumikomi Seigyo System no tame no Jikan Kudo Object shiko Software Kaihatsuho", The Transactions of the Institute of Electros, 2001, Vol.J84-D-1, No.4, pages 338 to 349 | 3 |
| Y | JP 2003-36176 A (Sony Corp.), 07 February, 2003 (07.02.03), Par. No. [0008] (Family: none) | 4 |

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not
considered to be of particular relevance

"E" earlier document but published on or after the international filing
date

"L" document which may throw doubts on priority claim(s) or which is
cited to establish the publication date of another citation or other
special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other
means

"P" document published prior to the international filing date but later
than the priority date claimed

"T" later document published after the international filing date or
priority date and not in conflict with the application but cited to
understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be
considered novel or cannot be considered to involve an inventive
step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be
considered to involve an inventive step when the document is
combined with one or more other such documents, such
combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
31 October, 2003 (31.10.03)

Date of mailing of the international search report
18 November, 2003 (18.11.03)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/09877

| C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT | | |
|---|--|-----------------------|
| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
| Y | JP 62-209626 A (Casio Computer Co., Ltd.), 14 September, 1987 (14.09.87), Page 1, lower right column, lines 1 to 8 (Family: none) | 11, 12 |
| A | JP 60-176145 A (Fujitsu Ltd.), 10 September, 1985 (10.09.85), Full text; all drawings (Family: none) | 1-12 |
| A | JP 9-26887 A (Mitsubishi Electric Corp.), 28 January, 1997 (28.01.97), Par. Nos. [0008] to [0015] (Family: none) | 1-12 |

A. 発明の属する分野の分類 (国際特許分類 (IPC))
Int. Cl⁷ G06F9/48, F02D45/00

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))
Int. Cl⁷ G06F9/46-9/54, F02D43/00-45/00

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1922-1996年
日本国公開実用新案公報 1971-2003年
日本国登録実用新案公報 1994-2003年
日本国実用新案登録公報 1996-2003年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

| 引用文献の カテゴリー* | 引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示 | 関連する 請求の範囲の番号 |
|-----------------|---|----------------------------|
| X Y | JP 2003-15885 A (キヤノン株式会社) 2003.01.17, 段落番号【0007】 (ファミリーなし) | 1, 2, 5-10 3, 4, 11, 12 |
| Y | 横山孝典, 外6名著, 組込み制御システムのための時間駆動オブジェクト指向ソフトウェア開発法, 電子情報通信学会論文誌, 2001, Vol. J84-D-1, No. 4, p. 338-349 | 3 |

☒ C欄の続きにも文献が列挙されている。

☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの
「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)
「O」 口頭による開示、使用、展示等に言及する文献
「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献
「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」 同一パテントファミリー文献

国際調査を完了した日

31.10.03

国際調査報告の発送日

18.11.03

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)
郵便番号100-8915
東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

羽立 章二



5B

2944

電話番号 03-3581-1101 内線 3545

| C (続き) . 関連すると認められる文献 | | |
|-----------------------|--|------------------|
| 引用文献の カテゴリー* | 引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示 | 関連する 請求の範囲の番号 |
| Y | J P 2003-36176 A (ソニー株式会社) 2003. 02. 07, 段落番号【0008】 (ファミリーなし) | 4 |
| Y | J P 62-209626 A (カシオ計算機株式会社) 1987. 09. 14, 第1頁右下欄第1行～第8行 (ファミリーなし) | 11, 12 |
| A | J P 60-176145 A (富士通株式会社) 1985. 09. 10, 全文, 全図 (ファミリーなし) | 1-12 |
| A | J P 9-26887 A (三菱電機株式会社) 1997. 01. 28, 段落番号【0008】 - 【0015】 (ファミリーなし) | 1-12 |